

10-06-00

A

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)*(Only for new nonprovisional applications under 37 CFR 1.53(b))*Docket No.
13768.172Total Pages in this Submission
5**TO THE ASSISTANT COMMISSIONER FOR PATENTS**Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

METHODS AND SYSTEMS FOR PERFORMING RELIABLE ASYNCHRONOUS NOTIFICATION OF HIGH-LEVEL DOCUMENT OPERATIONS

and invented by:

Jon B. Avner and Soner F. Terek

If a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 31 pages and including the following:
 - a. ☒ Descriptive Title of the Invention
 - b. ☐ Cross References to Related Applications *(if applicable)*
 - c. ☐ Statement Regarding Federally-sponsored Research/Development *(if applicable)*
 - d. ☐ Reference to Microfiche Appendix *(if applicable)*
 - e. ☒ Background of the Invention
 - f. ☒ Brief Summary of the Invention
 - g. ☒ Brief Description of the Drawings *(if drawings filed)*
 - h. ☒ Detailed Description
 - i. ☒ Claim(s) as Classified Below
 - j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
13768.172

Total Pages in this Submission
5

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*
- a. ☒ Formal Number of Sheets 6
- b. ☐ Informal Number of Sheets _____
4. ☒ Oath or Declaration
- a. ☒ Newly executed *(original or copy)* ☐ Unexecuted
- b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*
- c. ☒ With Power of Attorney ☐ Without Power of Attorney
- d. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application,
see 37 C.F.R. 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under
Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby
incorporated by reference therein.
6. ☐ Computer Program in Microfiche *(Appendix)*
7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*
- a. ☐ Paper Copy
- b. ☐ Computer Readable Copy *(identical to computer copy)*
- c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☒ Assignment Papers *(cover sheet & document(s))*
9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*
10. ☐ English Translation Document *(if applicable)*
11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Acknowledgment postcard
14. ☒ Certificate of Mailing
- ☐ First Class ☒ Express Mail *(Specify Label No.):* EL 550 340 297 US

**UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
13768.172

Total Pages in this Submission
5

Accompanying Application Parts (Continued)

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☒ Additional Enclosures *(please identify below):*

Attached sheet with correspondence address

Request That Application Not Be Published Pursuant To 35 U.S.C. 122(b)(2)

17. ☐ Pursuant to 35 U.S.C. 122(b)(2), Applicant hereby requests that this patent application not be published pursuant to 35 U.S.C. 122(b)(1). Applicant hereby certifies that the invention disclosed in this application has not and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication of applications 18 months after filing of the application.

Warning

An applicant who makes a request not to publish, but who subsequently files in a foreign country or under a multilateral international agreement specified in 35 U.S.C. 122(b)(2)(B)(i), must notify the Director of such filing not later than 45 days after the date of the filing of such foreign or international application. A failure of the applicant to provide such notice within the prescribed period shall result in the application being regarded as abandoned, unless it is shown to the satisfaction of the Director that the delay in submitting the notice was unintentional.

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
13768.172

Total Pages in this Submission
5

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	31	- 20 =	11	x \$18.00	\$198.00
Indep. Claims	4	- 3 =	1	x \$80.00	\$80.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$710.00
OTHER FEE (specify purpose) _____					\$0.00
TOTAL FILING FEE					\$988.00

- ☒ A check in the amount of **\$988.00** to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. _____ as described below. A duplicate copy of this sheet is enclosed.
- ☐ Charge the amount of _____ as filing fee.
- ☒ Credit any overpayment.
- ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).


Signature

ADRIAN J. LEE, Reg. # 42,785



022913

PATENT TRADEMARK OFFICE

Dated:

CC:

BACKGROUND OF THE INVENTION

1. The Field of the Invention

The present invention relates to the field of database management systems. In particular, the present invention relates to methods and systems for performing reliable asynchronous notification of high-level document operations in a database.

2. The Prior State of the Art

With the development of computer networks and the Internet, anyone who has access to an Internet capable computer may access information from all over the world. The present era has even been termed the “information age” due to the widespread abundance of information. Although this abundance of information is useful, individuals may easily be overwhelmed with information to the point where it is difficult to filter out relevant information from irrelevant information.

Database management systems are systems that manage collections of data (i.e., databases) so as to allow individuals, devices and applications to easily access, manage, and update information. Thus, database management systems provide a significant contribution to the information age by allowing for the efficient use of information.

Conventional database management systems such as the database management system 100 shown in Figure 1 include an underlying database 110 that contains organized data in the form of a number of tables such as table “A”, table “B”, table “C” and table “D”. Each table contains entries that associate documents with properties of the documents. For example, one table may contain a count of the number of files within given folders, another may list the memory size of given files, yet another may list other properties associated with given files, or folders. For each document, there is typically

1 more than one table that may associated with the document by, for example, describing
2 properties of the document.

3 A document (e.g., document 1, 2, 3 and 4 in Figure 1) is an identifiable entity from
4 the viewpoint of applications that use the database management system. For example, a
5 folder or an item such as an electronic mail message within the folder may be considered to
6 be a document since they are identified as entities from the viewpoint of the application
7 that uses the database management system. For example, in the context of electronic
8 messaging, an application may present folders such as "in-boxes" and "out-boxes" to a
9 user with corresponding electronic mail messages in the folders. These folders and
10 electronic mail messages are "documents" from the viewpoint of the application.
11 Similarly, files and directories within a file system may also be documents from the
12 viewpoint of the application that uses the database management system. The tables within
13 the underlying database are not considered to be documents since they are identified
14 internal to the database management system, and not at the higher level of the application
15 that uses the database management system.

16 The database application 120 generates high-level document commands (e.g., high-
17 level command 121) that relate to operations to be performed on a document. Examples of
18 such document commands might include operations such as, for example, move folder,
19 move message, delete message, copy folder, copy file, and so forth.

20 Each of these high-level document commands is received by the database engine
21 130 which implements the high-level document commands by executing a number of
22 object commands that result in objects such as tables being updated. Typically, the
23 database engine would use the disk access module 140 (e.g., a disk driver) of an operating
24 system to produce the physical control signals necessary to read and write the appropriate

The vast majority of the time, when a high-level document command having certain criteria is issued, the subscribing client applications are properly notified of the high-level document command and thus can respond appropriately. However, sometimes computer systems become inoperative due to system failures, corruption, power loss, or the like. In such cases, there is a possibility that the high-level document command is implemented in the database, but that the subscribing client application never receives notification of the implementation. For example, the client application may become inoperative and thus become unable to receive and interpret a notification that a certain high-level document command has been implemented. Also, the database management

1 system itself may become inoperative so that although the command has been
2 implemented, no notification was ever dispatched.

3 Thus, conventional system contain no guarantee that the proper notifications of a
4 high-level document command will always be dispatched or received if a high-level
5 document command is implemented. Therefore, what are desired are methods and systems
6 for improving accuracy and reliability in notifying client applications that certain high-
7 level document commands have been implemented in a database.

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

OFFICE " 5400360

SUMMARY OF THE INVENTION

The principles of the present invention provide for the reliable asynchronous notification to client applications of the implementation of predefined high-level document commands that are implemented in a database managed by a database management system. These notifications are "asynchronous" since the high-level document commands are actually implemented in the database prior to sending the notification to client applications. Thus, the notified client applications do not affect the implementation of the high-level document commands in the database.

The database management system is issued a number of high-level document commands which are to be implemented in the underlying database. Such high-level document commands include high-level commands that relate to actions to be taken on documents such as folders, files, messages and other entities that are identified at the level of the application that uses the database management system. For example, in electronic messaging applications, one high-level document command may be to add a message to a folder.

Ultimately, the database engine implements the high-level document command by altering one or more tables in the database. However, before implementing the high-level command, an entry is set in a notification table stored in a persistent memory such as the database. Alternatively, the entry in the notification table is set in the same transaction as any other table changes that are made to implement the high-level command. In this transaction, either all actions in the transaction are performed (all relevant tables are altered) or no actions in the transaction are taken (none of the relevant tables are altered). Thus, if the entry in the notification table is set, then the corresponding high-level document command is guaranteed to have been implemented in the database.

1 Ideally, the client application that requested notification and the database
2 management system will continue to be operational. In this ideal case, the notification that
3 the given high-level document command has been implemented is dispatched to the client
4 application. The client application then acknowledges to the database management system
5 that the notification has been received. In response, the entry is deleted in the notification
6 table in the database. The notification may be resent if the database management system
7 does not receive such acknowledgement within an expected time period.

8 To make sure that the client management system properly performs all actions
9 responsive to the notification, the entry in the notification table may remain set until the
10 database management system receives acknowledgement from the client application that
11 all necessary actions have been performed by the client application. Thus, if a client
12 application fails while implementing any necessary actions responsive to the notification,
13 the notification will be resent to allow the client application to re-perform any necessary
14 actions once the client application becomes operational again.

15 Suppose now that a system failure occurs at the database management system after
16 the high-level document command is implemented in the database, but before notification
17 is dispatched. Upon restarting the database management system, the system will check the
18 notification table to see what notifications entries are set. These entries correspond to
19 notifications for which the database management system has not received
20 acknowledgement from the client application. The database management system then
21 sends corresponding notifications to the client application. In this manner, the client
22 application receives notification even if a database management system failure occurs.

23 Now suppose that the notification is indeed dispatched, but that the client
24 application has failed so as to be unable to receive the notification. The database

1 management system will not receive acknowledgement from the client application.
2 Therefore, the database application will resend the notification until the notification is
3 acknowledged upon restarting the client application. In this manner, the client application
4 receives notification even if a client application failure occurs.

5 Therefore, the principles of the present invention provide for a reliable way of
6 notifying client application when particular high-level document commands are
7 implemented in a database, the notifications surviving even failure of the database
8 management system or client application.

9 Additional features and advantages of the invention will be set forth in the
10 description which follows, and in part will be obvious from the description, or may be
11 learned by the practice of the invention. The features and advantages of the invention may
12 be realized and obtained by means of the instruments and combinations particularly
13 pointed out in the appended claims. These and other features of the present invention will
14 become more fully apparent from the following description and appended claims, or may
15 be learned by the practice of the invention as set forth hereinafter.

004001" 54008950

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

Figure 1 consists of 12 micrographs arranged vertically, showing the stages of chick embryo development. The stages are labeled on the right side of each image: 1. Fertilized egg, 2. 2-cell stage, 3. 4-cell stage, 4. 8-cell stage, 5. Morula stage, 6. Gastrula stage, 7. Head stage, 8. Heart stage, 9. Somite stage, 10. Limb stage, 11. Hatching stage, and 12. Hatched chick. The images show the progression from a single cell to a fully formed chick in the egg.

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

1 structures stored thereon. Such computer-readable media can be any available media
2 which can be accessed by a general purpose or special purpose computer. By way of
3 example, and not limitation, such computer-readable media can comprise physical storage
4 media such as RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic
5 disk storage or other magnetic storage devices, or any other medium which can be used to
6 carry or store desired program code means in the form of computer-executable instructions
7 or data structures and which can be accessed by a general purpose or special purpose
8 computer.

9 When information is transferred or provided over a network or another
10 communications connection (either hardwired, wireless, or a combination of hardwired or
11 wireless) to a computer, the computer properly views the connection as a computer-
12 readable medium. Thus, any such a connection is properly termed a computer-readable
13 medium. Combinations of the above should also be included within the scope of
14 computer-readable media. Computer-executable instructions comprise, for example,
15 instructions and data which cause a general purpose computer, special purpose computer,
16 or special purpose processing device to perform a certain function or group of functions.

17 Figure 3 and the following discussion are intended to provide a brief, general
18 description of a suitable computing environment in which the invention may be
19 implemented. Although not required, the invention will be described in the general context
20 of computer-executable instructions, such as program modules, being executed by
21 computers in network environments. Generally, program modules include routines,
22 programs, objects, components, data structures, etc. that perform particular tasks or
23 implement particular abstract data types. Computer-executable instructions, associated
24 data structures, and program modules represent examples of the program code means for

With reference to Figure 3, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional computer 320, including a processing unit 321, a system memory 322, and a system bus 323 that couples various system components including the system memory 322 to the processing unit 321. The system bus 323 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 324 and random access memory (RAM) 325. A basic input/output system (BIOS) 326, containing the basic routines that help transfer information between elements within the computer 320, such as during start-up, may be stored in ROM 324.

1 The computer 320 may also include a magnetic hard disk drive 327 for reading
2 from and writing to a magnetic hard disk 339, a magnetic disk drive 328 for reading from
3 or writing to a removable magnetic disk 329, and an optical disk drive 330 for reading
4 from or writing to removable optical disk 331 such as a CD-ROM or other optical media.
5 The magnetic hard disk drive 327, magnetic disk drive 328, and optical disk drive 330 are
6 connected to the system bus 323 by a hard disk drive interface 332, a magnetic disk drive-
7 interface 333, and an optical drive interface 334, respectively. The drives and their
8 associated computer-readable media provide nonvolatile storage of computer-executable
9 instructions, data structures, program modules and other data for the computer 320.
10 Although the exemplary environment described herein employs a magnetic hard disk 339,
11 a removable magnetic disk 329 and a removable optical disk 331, other types of computer
12 readable media for storing data can be used, including magnetic cassettes, flash memory
13 cards, digital video disks, Bernoulli cartridges, RAMs, ROMs, and the like.

14 Program code means comprising one or more program modules may be stored on
15 the hard disk 339, magnetic disk 329, optical disk 331, ROM 324 or RAM 325, including
16 an operating system 335, one or more application programs 336, other program modules
17 337, and program data 338. A user may enter commands and information into the
18 computer 320 through keyboard 340, pointing device 342, or other input devices (not
19 shown), such as a microphone, joy stick, game pad, satellite dish, scanner, or the like.
20 These and other input devices are often connected to the processing unit 321 through a
21 serial port interface 346 coupled to system bus 323. Alternatively, the input devices may
22 be connected by other interfaces, such as a parallel port, a game port or a universal serial
23 bus (USB). A monitor 347 or another display device is also connected to system bus 323
24 via an interface, such as video adapter 348. In addition to the monitor, personal computers

1 typically include other peripheral output devices (not shown), such as speakers and
2 printers.

3 The computer 320 may operate in a networked environment using logical
4 connections to one or more remote computers, such as remote computers 349a and 349b.
5 Remote computers 349a and 349b may each be another personal computer, a server, a
6 router, a network PC, a peer device or other common network node, and typically include
7 many or all of the elements described above relative to the computer 320, although only
8 memory storage devices 350a and 350b and their associated application programs 336a and
9 336b have been illustrated in Figure 3. The logical connections depicted in Figure 3
10 include a local area network (LAN) 351 and a wide area network (WAN) 352 that are
11 presented here by way of example and not limitation. Such networking environments are
12 commonplace in office-wide or enterprise-wide computer networks, intranets and the
13 Internet.

14 When used in a LAN networking environment, the computer 320 is connected to
15 the local network 351 through a network interface or adapter 353. When used in a WAN
16 networking environment, the computer 320 may include a modem 354, a wireless link, or
17 other means for establishing communications over the wide area network 352, such as the
18 Internet. The modem 354, which may be internal or external, is connected to the system
19 bus 323 via the serial port interface 346. In a networked environment, program modules
20 depicted relative to the computer 320, or portions thereof, may be stored in the remote
21 memory storage device. It will be appreciated that the network connections shown are
22 exemplary and other means of establishing communications over wide area network 352
23 may be used.
24

1 Figure 4 illustrates a database management system 400 in accordance with the
2 present invention. Although, not required, the database management system 400 may be
3 implemented in the computing environment shown in Figure 4. In contrast to the prior art
4 database management system shown in Figure 1, the database management system 400
5 includes a notification table 450. The operation of the database management system 400
6 of Figure 4 will be described with respect to the method 500 illustrates in Figure 5.

7 Figure 5 shows a flowchart of a method performed by the database management
8 system 400 to perform reliable notification when a high-level document command is
9 implemented by the database management system. The database application 420 issues a
10 high-level document command 421 to the database engine 430. This high-level document
11 command is for performing a function on a document that the database application
12 recognizes to be a separately identifiable entity. For example, if the database management
13 system supports electronic mail applications, the document may be an electronic mail
14 message or a folder that contains the electronic mail message. Typical operations might be
15 to move, delete, copy, update, or add a new document.

16 As the database engine 430 receives each high-level document command, the
17 database engine implements the high-level document command in the database 410 (act
18 501). Typically, implementing a high-level document command in the database may
19 involve changing entries in a number of tables such as tables "A", "B", "C" and "D" since
20 each table could potentially have an entry for the corresponding document. The client
21 application that uses the database management system 400 is typically not concerned with
22 the details of which table needs to be changed. The database engine implements those
23 details usually unbeknownst to the client application that generated the high-level
24 document command. Highly reliable database engines typically perform changes to the

1 having subscribed or perhaps a third party client application has directed the other client
2 application to receive notifications of the implementation of certain high-level document
3 commands.

4 Once the client applications to be notified are identified (act 503), a notification
5 that the high-level document command has been implemented in then dispatched to the
6 identified client applications (act 504). Messaging may be employed to dispatch the
7 notification as when the database management system and the client application to be
8 notified do not reside on the same machine. Alternatively, function calls through an
9 application program interface or through other interfaces may be employed when the
10 database management system and the client application reside on the same machine.

11 Once notified, the client application may then perform any number and variety of
12 processes in response to receiving the notification. For example, if the original client
13 application has added a new electronic mail message to a public folder, each of the client
14 applications having access to the public folder may be notified. If the client application is
15 already displaying the public folder, the client application would then refresh a display to
16 reflect the addition.

17 Ultimately, the database management system receives acknowledgement back from
18 the client application that received the notification (act 505). The client application may
19 return this acknowledgement as soon as the client application receives notification.
20 However, it may often be beneficial to return acknowledgement well after this time as soon
21 as the client application has completed performing any processes in response to receiving
22 notifications. This ensures that the client application has successfully completed its
23 purpose in being notified in the first place.
24

1 Once the database management system receives acknowledgement back from the
2 client application (act 505), the database management system alters the notification table
3 450 to delete reference to the implementation of the high-level document command (act
4 506). Thus, the notification table reflects the implementation of every high-level document
5 command that has been implemented in the database, but for which the database
6 management system has not yet received acknowledgement back from the client
7 application. In this way, the notification table reflects those notifications that should have
8 been sent to the client application, but for which the notification success is uncertain.

9 As will now be described, this method permits for notifications of the
10 implementation of high-level document commands to survive even system failures. For
11 example, suppose that, for whatever reason, the high-level document command is
12 implemented in the database, but the appropriate notification is never dispatched. Perhaps
13 the database management system experiences a failure after the high-level document
14 command was implemented, but before the database management system actually sent the
15 notification. In this case, at least the database management system would have set the
16 appropriate entries in the notification table to reflect that the high-level document
17 command was implemented.

18 Figure 6 illustrates a flowchart of method 600 of how the database management
19 system would recover from such a failure and send the appropriate notification. Upon
20 restarting the database management system (act 601), the database management system
21 checks the notification table to identify any high-level document commands that were
22 implemented, but for which there has not yet been acknowledgement back from the client
23 application (act 602). If there are none (NO in decision block 602), then no notification
24 needs to be resent and the method ends. If there are some (YES in decision block 602),

1 then the subscribers to the notification are identified (act 603), and the appropriate client
2 applications are notified (act 604). Upon receiving acknowledgement back from the client
3 applications (act 605), the appropriate entry in the notification table is cleared (act 606).
4 Thus, even a failure of the database management system will not prevent appropriate
5 notification.

6 Furthermore, if, for some reason, the notification was sent, but the client
7 application never received notification, the client application will still be notified. For
8 example, suppose that the machine hosting the client application was to fail, or perhaps the
9 notification medium (e.g., a network) was to fail resulting in the client application not
10 receiving the notification. In this case, the client application would not return an
11 acknowledgement and the entry would remain in the notification table. The database
12 management system may be configured to continue to send the notification on a periodic
13 basis absent an acknowledgement until acknowledgement is finally returned. Thus,
14 notification may even be made if the client application or network should fail.

15 Thus, the principles of the present invention provide for a reliable way of notifying
16 client applications when a particular high-level document command is implemented in a
17 database. The present invention may be embodied in other specific forms without
18 departing from its spirit or essential characteristics. The described embodiments are to be
19 considered in all respects only as illustrative and not restrictive. The scope of the
20 invention is, therefore, indicated by the appended claims rather than by the foregoing
21 description. All changes which come within the meaning and range of equivalency of the
22 claims are to be embraced within their scope.

23 What is claimed and desired to be secured by United States Letters Patent is:
24

1 1. In a database management system that includes a database engine that
2 accesses and updates objects in a database, the database engine receiving high-level
3 document commands, each high-level document command for performing an operation on
4 a document that is associated with a plurality of tables in the database, a method for
5 reliably notifying client applications of the implementation of particular high-level
6 document commands in the database, the notification surviving even system failures, the
7 method comprising the following:

8 an act of implementing a high-level document command in the database;

9 an act of altering a persistently stored notification table to reflect the
10 implementation of the high-level document command in the database;

11 an act of identifying one or more client applications that are to be notified
12 of the implementation of the high-level document command;

13 an act of dispatching a notification of the implementation to the one or more
14 identified client applications;

15 an act of receiving acknowledgement from the one or more client
16 applications that the notification has been received; and

17 in response to receiving acknowledgement, an act of altering the
18 notification table to reflect that the client application no longer needs to be notified
19 of the implementation of the high-level document command in the database.

20
21 2. The method in accordance with Claim 1, wherein the notification table is
22 stored in the database.
23
24

1 3. The method in accordance with Claim 1, wherein the act of dispatching a
2 notification of the implementation to the one or more identified client applications
3 comprises an act of transmitting a message to a machine that hosts the client application,
4 the machine that host the client application being different than the machine that hosts the
5 database management system.

6
7 4. The method in accordance with Claim 1, wherein the act of dispatching a
8 notification of the implementation to the one or more identified client applications
9 comprises an act of passing the notification through a function call to the identified client
10 application, the client application hosted by the same machine as at least the portion of the
11 database management system responsible for performing the act of dispatching the
12 notification.

13
14 5. The method in accordance with Claim 1, wherein the act of implementing a
15 high-level document command in the database and the act of altering a persistently stored
16 notification table to reflect the implementation of the high-level document command in the
17 database are atomically performed, the acts of implementing and altering either both
18 occurring or both not occurring at all.

19
20 6. The method in accordance with Claim 1, wherein the act of implementing a
21 high-level document command in the database and the act of altering a persistently stored
22 notification table to reflect the implementation occur in the same transaction of a database
23 engine.
24

1 7. The method in accordance with Claim 1, wherein the document comprises
2 an electronic mail message.

3
4 8. The method in accordance with Claim 1, wherein the document comprises a
5 folder.

6
7 9. The method in accordance with Claim 1, wherein the act of implementing
8 the high-level document command in the database comprises an act of moving the
9 document.

10
11 10. The method in accordance with Claim 1, wherein the act of implementing
12 the high-level document command in the database comprises an act of deleting the
13 document.

14
15 11. The method in accordance with Claim 1, wherein the act of implementing
16 the high-level document command in the database comprises an act of copying the
17 document.

18
19 12. The method in accordance with Claim 1, wherein the act of implementing
20 the high-level document command in the database comprises an act of adding the
21 document.

1 13. The method in accordance with Claim 1, wherein the act of implementing
2 the high-level document command in the database comprises an act of updating the
3 document.

4
5 14. The method in accordance with Claim 1, wherein the received
6 acknowledgement indicates that the client application has received the notification.

7
8 15. The method in accordance with Claim 1, wherein the received
9 acknowledgement indicates that the client application has implemented processes in
10 response to the notification.

1 16. In a database management system that includes a database engine that
2 accesses and updates objects in a database, the database engine receiving high-level
3 document commands, each high-level document command for performing an operation on
4 a document that is associated with a plurality of tables in the database, a method for
5 reliably notifying client applications of the implementation of particular high-level
6 document commands in the database, the notification surviving even system failures, the
7 method comprising the following:

8 an act of implementing a high-level document command in the database;

9 and

10 a step for ensuring a corresponding notification about the high-level
11 document command is preserved until all of the client applications to be notified
12 acknowledge at least receipt of the notification.

13
14 17. The method in accordance with Claim 16, wherein the step for ensuring a
15 corresponding notification about the high-level document command is preserved comprises
16 the following:

17 an act of altering a persistently stored notification table to reflect the
18 implementation of the high-level document command in the database;

19 an act of identifying one or more client applications that are to be notified
20 of the implementation of the high-level document command;

21 an act of dispatching a notification of the implementation to the one or more
22 identified client applications;

23 an act of receiving acknowledgement from the one or more client
24 applications that the notification has been received; and

1 in response to receiving acknowledgement, an act of altering the
2 notification table to reflect that the client application no longer needs to be notified
3 of the implementation of the high-level document command in the database.
4

5 18. The method in accordance with Claim 16, wherein the notification table is
6 stored in the database.
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

19. A computer program product for use in a database management system that includes a database engine that accesses and updates objects in a database, the database engine receiving high-level document commands, each high-level document command for performing an operation on a document that is associated with a plurality of tables in the database, the computer-program product for implementing a method for reliably notifying client applications of the implementation of particular high-level document commands in the database, the notification surviving even system failures, the computer program product including a computer-readable medium having stored thereon computer-executable instructions for performing the following:

an act of causing a high-level document command to be implemented in the database;

an act of causing a persistently stored notification table to be altered to reflect the implementation of the high-level document command in the database;

an act of identifying one or more client applications that are to be notified of the implementation of the high-level document command;

an act of causing a notification of the implementation to be dispatched to the one or more identified client applications;

an act of detecting receipt of acknowledgement from the one or more client applications that the notification has been received; and

in response to receiving acknowledgement, an act of causing the notification table to be altered reflect that the client application no longer needs to be notified of the implementation of the high-level document command in the database.

1 20. The computer program product in accordance with Claim 19, wherein the
2 computer-executable instructions for performing the act of causing a notification of the
3 implementation to be dispatched to the one or more identified client applications comprise
4 computer-executable instruction for performing an act of causing a message to be
5 transmitted to a machine that hosts the client application, the machine that hosts the client
6 application being different than the machine that hosts the database management system.

7
8 21. The computer program product in accordance with Claim 19, wherein the
9 computer-executable instructions for performing the act of causing a notification of the
10 implementation to be dispatched to the one or more identified client applications comprises
11 computer-executable instructions for performing an act of causing the notification to be
12 passed through a function call to the identified client application, the client application
13 hosted by the same machine as at least the portion of the database management system
14 responsible for performing the act of dispatching the notification.

15
16 22. The computer program product in accordance with Claim 19, wherein the
17 computer-executable instructions for performing the act of implementing a high-level
18 document command in the database and the computer-executable instructions for
19 performing the act of altering a persistently stored notification table to reflect the
20 implementation of the high-level document command in the database are configured to
21 atomically performing the acts of implementing and the act of altering.

22
23 23. The computer program product in accordance with Claim 19, wherein the
24 document comprises an electronic mail message.

1
2 24. The computer program product in accordance with Claim 19, wherein the
3 document comprises a folder.
4

5 25. The computer program product in accordance with Claim 19, wherein the
6 computer-executable instructions for performing the act of causing the high-level
7 document command to be implemented in the database comprises comprise computer-
8 executable instructions for performing an act of moving the document.
9

10 26. The computer program product in accordance with Claim 19, wherein the
11 computer-executable instructions for performing the act of causing the high-level
12 document command to be implemented in the database comprises comprise computer-
13 executable instructions for performing an act of deleting the document.
14

15 27. The computer program product in accordance with Claim 19, wherein the
16 computer-executable instructions for performing the act of causing the high-level
17 document command to be implemented in the database comprises comprise computer-
18 executable instructions for performing an act of copying the document.
19

20 28. The computer program product in accordance with Claim 19, wherein the
21 computer-executable instructions for performing the act of causing the high-level
22 document command to be implemented in the database comprises comprise computer-
23 executable instructions for performing an act of adding the document.
24

1 30. In a database management system that includes a database engine that
2 accesses and updates tables in a database, the database engine receiving high-level
3 document commands, each high-level document command for performing an operation on
4 a document that is associated with a plurality of tables in the database, a method for
5 recovering from a database management system failure while allowing notifications that
6 were to be sent but for the failure to be sent to one or more client applications upon
7 restarting the database management system, the method comprising the following:

8 an act of examining a notification table upon restarting the database
9 management system, the notification table identifying implemented high-level
10 document commands for which one or more client applications should be notified;

11 based on the examination of the notification table, an act of identifying a
12 notification that should have been sent to the one or more client applications, but
13 for which acknowledgement has not been received from the one or more client
14 applications; and

15 an act of dispatching the notification of the implementation to the one or
16 more identified client applications.

17
18 31. The method in accordance with Claim 30, further comprising the following:

19 an act of receiving acknowledgement from the one or more client
20 applications that the notification has been received; and

21 in response to receiving acknowledgement, an act of altering the
22 notification table to reflect that the client application no longer needs to be notified
23 of the implementation of the high-level document command in the database.
24

G:\DATA\PAT\WORDPAT\13768.172.doc

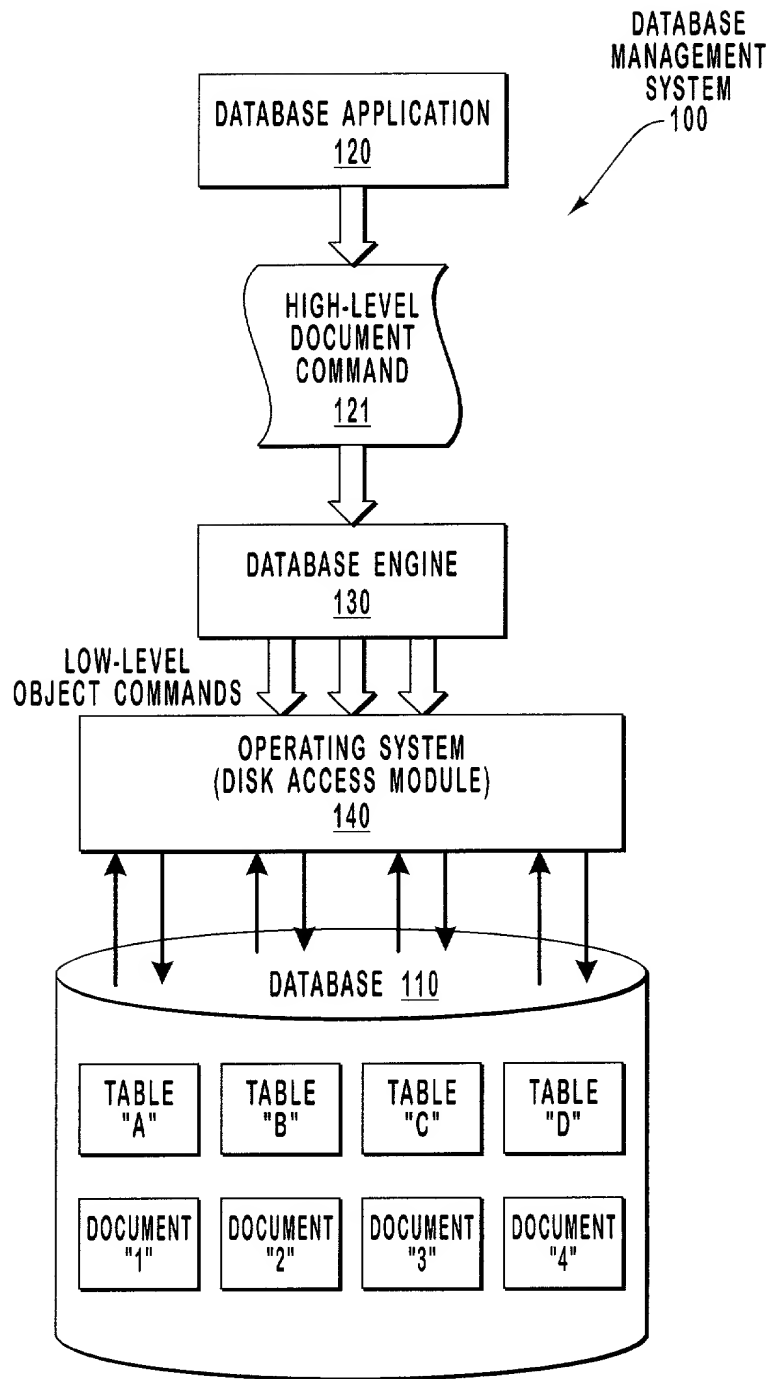


FIG. 1
(PRIOR ART)

004001" 54002950

METHOD
200

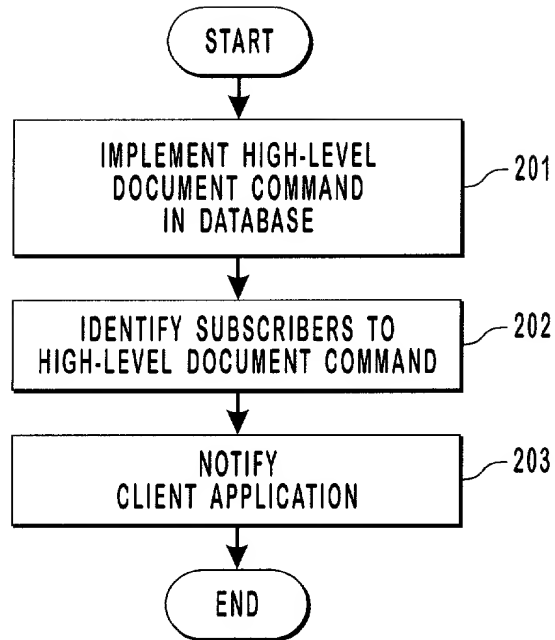
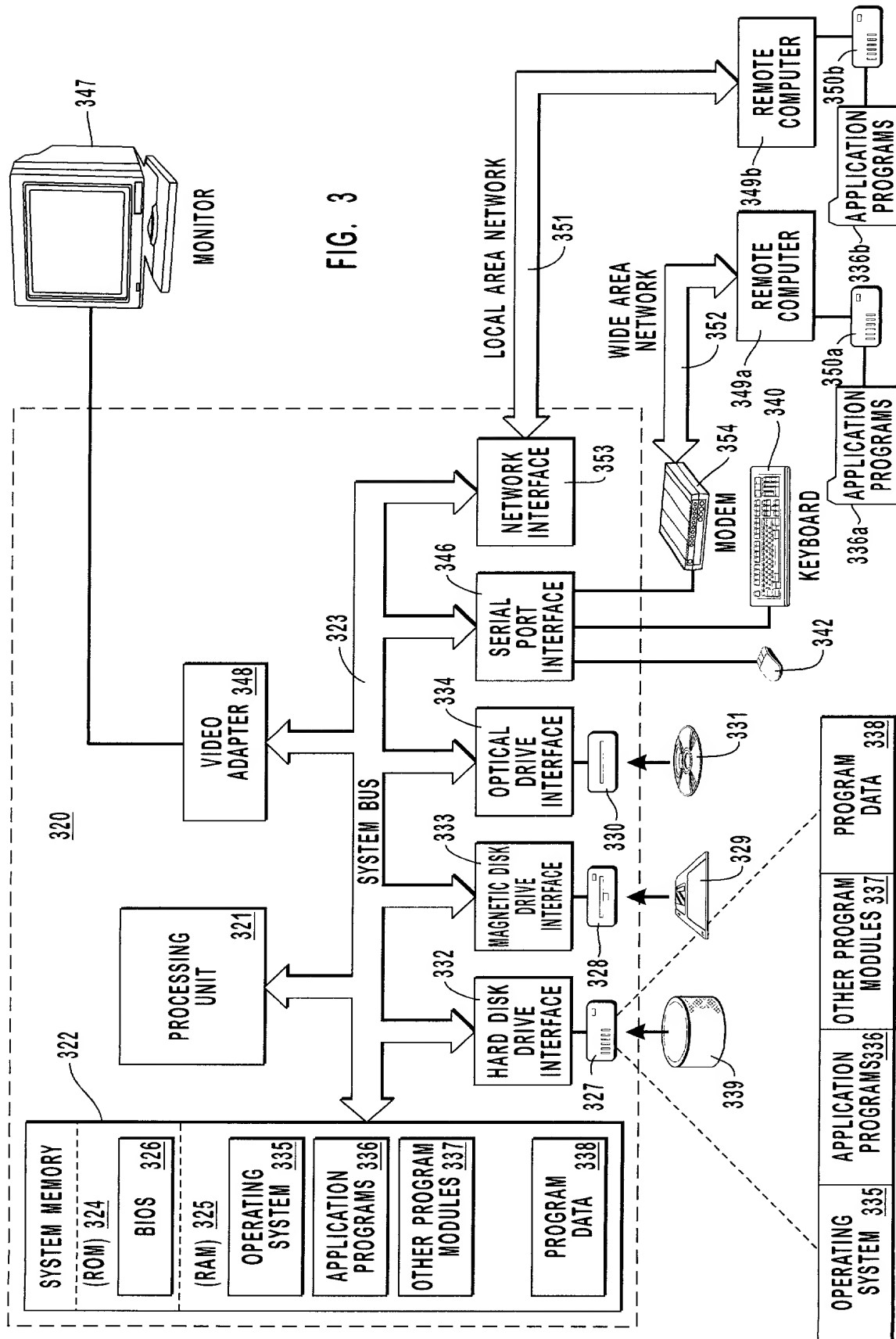


FIG. 2
(PRIOR ART)



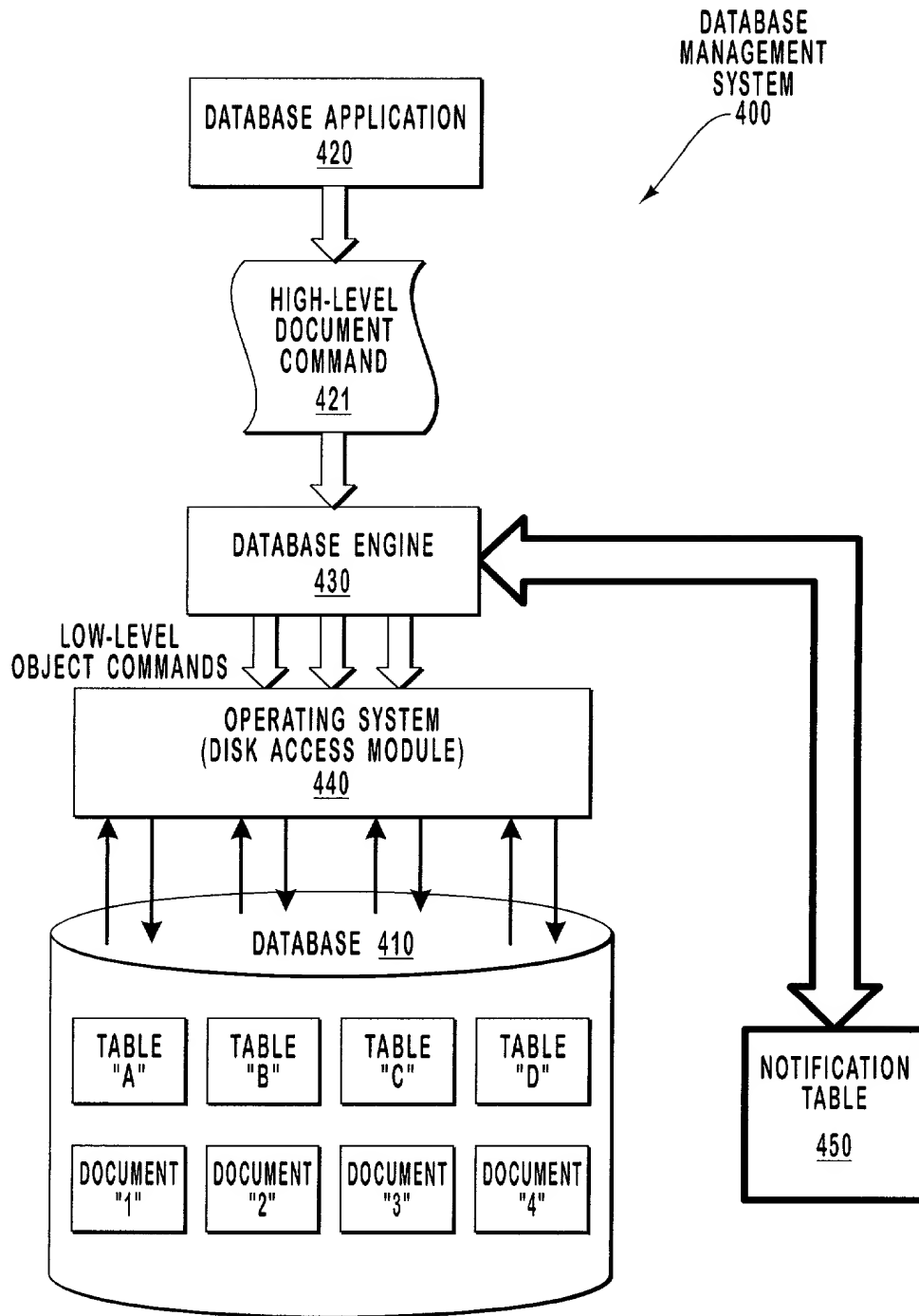


FIG. 4

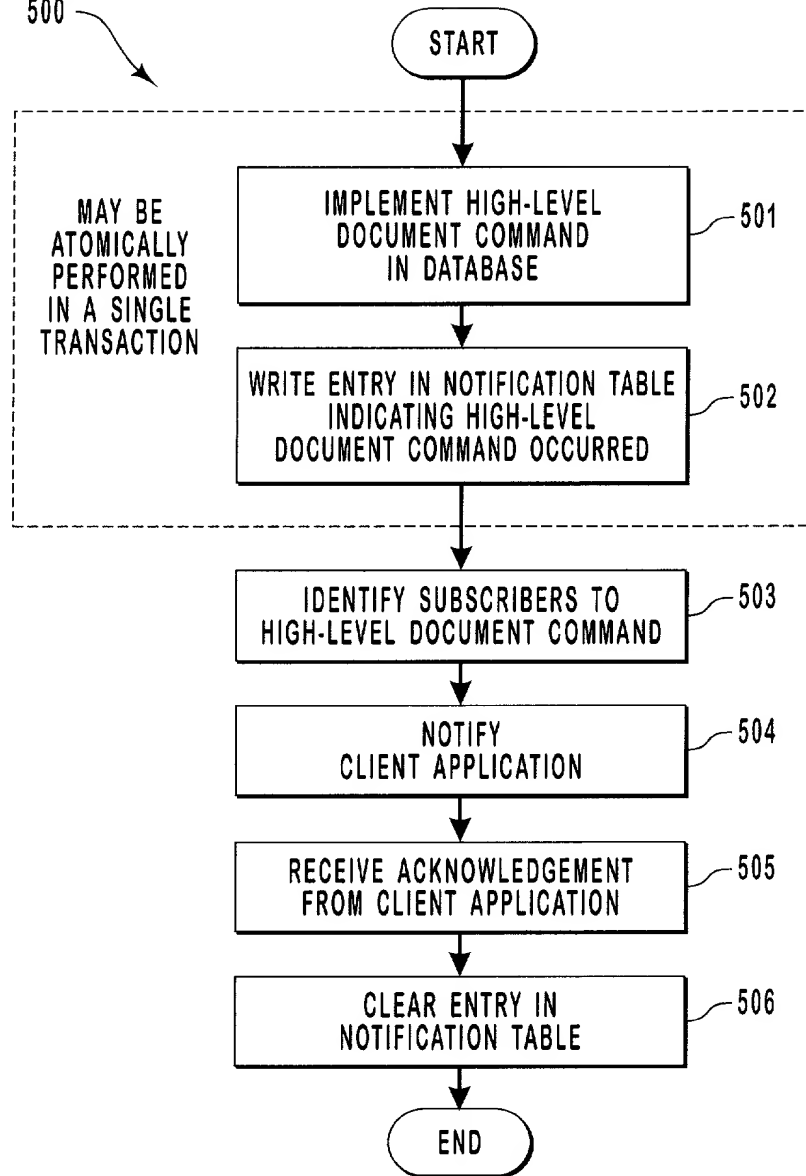
METHOD
500

FIG. 5

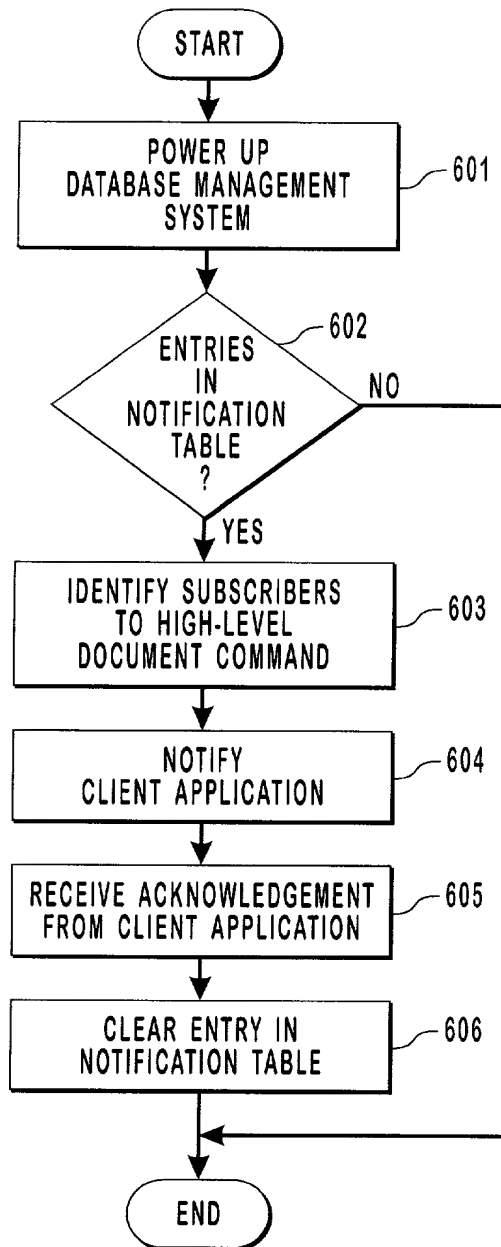
METHOD
600

FIG. 6

Express Mailing Label No: EL 550 340 297 US

PATENT APPLICATION

Docket No: 13768.172

DECLARATION, POWER OF ATTORNEY, AND PETITION

I, Jon B. Avner, declare: that I am a citizen of the United States, that my residence and post office address is 17234 SE 40th Place, Bellevue, Washington 98008; and I, Soner F. Terek, declare: that I am a citizen of Turkey, that my residence and post office address is 13695 NE 32nd Place, Bellevue, Washington 98005; that we verily believe we are the original, first, and joint inventors of the subject matter of the invention or discovery entitled "METHODS AND SYSTEMS FOR PERFORMING RELIABLE ASYNCHRONOUS NOTIFICATION OF HIGH-LEVEL DOCUMENT OPERATIONS," for which a patent is sought

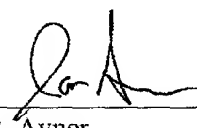
and which is described and claimed in the specification attached hereto;

and that we acknowledge the duty to disclose information which is material to the examination of this application in accordance with Section 1.56(a) of Title 37 of the Code of Federal Regulations.

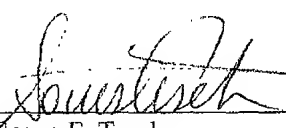
We declare further that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful, false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful, false statements may jeopardize the validity of the application or any patent issuing thereon.

We hereby appoint as our attorneys and/or patent agents all listed under Customer No. 022913; and DANIEL D. CROUSE, Registration No. 32,022; and KATIE SAKO, Registration No. 32,628, of MICROSOFT CORPORATION, One Microsoft Way, Redmond, Washington 98052,

Signed at Redmond, Washington this 4 day of October, 2000.

Inventor: 
Jon B. Avner
17234 SE 40th Place
Bellevue, Washington 98008

Signed at Redmond, Washington this 4th day of October, 2000.

Inventor: 
Soner F. Terek
13695 NE 32nd Place
Bellevue, Wahington 98005

G:\DATA\WPDOCSRN\MICROSOFT\OTHERDOC\1002 DEC-JOINT_172.DOC

004001" 54002560